;; Auto-generated. Do not edit!

(when (boundp 'robotnik\_msgs\_new::MotorPID)

(if (not (find-package "ROBOTNIK\_MSGS\_NEW"))

(make-package "ROBOTNIK\_MSGS\_NEW"))

(shadow 'MotorPID (find-package "ROBOTNIK\_MSGS\_NEW")))

(unless (find-package "ROBOTNIK\_MSGS\_NEW::MOTORPID")

(make-package "ROBOTNIK\_MSGS\_NEW::MOTORPID"))

(in-package "ROS")

;;//! \htmlinclude MotorPID.msg.html

(defclass robotnik\_msgs\_new::MotorPID

:super ros::object

:slots (\_can\_id \_name \_kp \_ki \_kd ))

(defmethod robotnik\_msgs\_new::MotorPID

(:init

(&key

((:can\_id \_\_can\_id) (make-array 0 :initial-element 0 :element-type :integer))

((:name \_\_name) (let (r) (dotimes (i 0) (push "" r)) r))

((:kp \_\_kp) (make-array 0 :initial-element 0.0 :element-type :float))

((:ki \_\_ki) (make-array 0 :initial-element 0.0 :element-type :float))

((:kd \_\_kd) (make-array 0 :initial-element 0.0 :element-type :float))

)

(send-super :init)

(setq \_can\_id \_\_can\_id)

(setq \_name \_\_name)

(setq \_kp \_\_kp)

(setq \_ki \_\_ki)

(setq \_kd \_\_kd)

self)

(:can\_id

(&optional \_\_can\_id)

(if \_\_can\_id (setq \_can\_id \_\_can\_id)) \_can\_id)

(:name

(&optional \_\_name)

(if \_\_name (setq \_name \_\_name)) \_name)

(:kp

(&optional \_\_kp)

(if \_\_kp (setq \_kp \_\_kp)) \_kp)

(:ki

(&optional \_\_ki)

(if \_\_ki (setq \_ki \_\_ki)) \_ki)

(:kd

(&optional \_\_kd)

(if \_\_kd (setq \_kd \_\_kd)) \_kd)

(:serialization-length

()

(+

;; int32[] \_can\_id

(\* 4 (length \_can\_id)) 4

;; string[] \_name

(apply #'+ (mapcar #'(lambda (x) (+ 4 (length x))) \_name)) 4

;; float32[] \_kp

(\* 4 (length \_kp)) 4

;; float32[] \_ki

(\* 4 (length \_ki)) 4

;; float32[] \_kd

(\* 4 (length \_kd)) 4

))

(:serialize

(&optional strm)

(let ((s (if strm strm

(make-string-output-stream (send self :serialization-length)))))

;; int32[] \_can\_id

(write-long (length \_can\_id) s)

(dotimes (i (length \_can\_id))

(write-long (elt \_can\_id i) s)

)

;; string[] \_name

(write-long (length \_name) s)

(dolist (elem \_name)

(write-long (length elem) s) (princ elem s)

)

;; float32[] \_kp

(write-long (length \_kp) s)

(dotimes (i (length \_kp))

(sys::poke (elt \_kp i) (send s :buffer) (send s :count) :float) (incf (stream-count s) 4)

)

;; float32[] \_ki

(write-long (length \_ki) s)

(dotimes (i (length \_ki))

(sys::poke (elt \_ki i) (send s :buffer) (send s :count) :float) (incf (stream-count s) 4)

)

;; float32[] \_kd

(write-long (length \_kd) s)

(dotimes (i (length \_kd))

(sys::poke (elt \_kd i) (send s :buffer) (send s :count) :float) (incf (stream-count s) 4)

)

;;

(if (null strm) (get-output-stream-string s))))

(:deserialize

(buf &optional (ptr- 0))

;; int32[] \_can\_id

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_can\_id (instantiate integer-vector n))

(dotimes (i n)

(setf (elt \_can\_id i) (sys::peek buf ptr- :integer)) (incf ptr- 4)

))

;; string[] \_name

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_name (make-list n))

(dotimes (i n)

(let (n) (setq n (sys::peek buf ptr- :integer)) (incf ptr- 4) (setf (elt \_name i) (subseq buf ptr- (+ ptr- n))) (incf ptr- n))

))

;; float32[] \_kp

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_kp (instantiate float-vector n))

(dotimes (i n)

(setf (elt \_kp i) (sys::peek buf ptr- :float)) (incf ptr- 4)

))

;; float32[] \_ki

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_ki (instantiate float-vector n))

(dotimes (i n)

(setf (elt \_ki i) (sys::peek buf ptr- :float)) (incf ptr- 4)

))

;; float32[] \_kd

(let (n)

(setq n (sys::peek buf ptr- :integer)) (incf ptr- 4)

(setq \_kd (instantiate float-vector n))

(dotimes (i n)

(setf (elt \_kd i) (sys::peek buf ptr- :float)) (incf ptr- 4)

))

;;

self)

)

(setf (get robotnik\_msgs\_new::MotorPID :md5sum-) "a4f1747645e7d598483fc2ed471a485d")

(setf (get robotnik\_msgs\_new::MotorPID :datatype-) "robotnik\_msgs\_new/MotorPID")

(setf (get robotnik\_msgs\_new::MotorPID :definition-)

"# either can\_id or name are set

# if can\_id is -1, then this refers to all motors.

int32[] can\_id

string[] name

float32[] kp

float32[] ki

float32[] kd

")

(provide :robotnik\_msgs\_new/MotorPID "a4f1747645e7d598483fc2ed471a485d")